



**JOINT INSTITUTE FOR
NUCLEAR RESEARCH**

*Laboratory Of Information
Technologies*

**FINAL REPORT ON THE
INTEREST PROGRAM**

INTRODUCTION TO QUANTUM COMPUTING

Supervisor : Dr Mihai Dima

Student : Aarsh Chaube , India

SRM Institute Of Science and Technology, Chennai

Participation Date -: 24th May - 2nd July, 2021

Dubna, 2021

Abstract

The goal of this project was to explore Quantum Computing from ground up. We started off with Quantum Mechanics and essential Hardware, and then proceeded to qubit measurement experiments. Furthermore, we looked at various quantum gates, states and explored Grover's Algorithm in a bit more detail. We utilized CPP physics packages , namely SU2 and CPX for performing the measurements, on the HYBRILIT supercomputing platform, and plotted the results using CERN's ROOT. At last, we delved into IBM's qiskit for coding other quantum circuits and various Algorithms .I've also gone beyond the scope of the project to review some more ideas in the field and have marked the content as [**Optional**] in this report

Tasks

1. Understand two state Quantum Mechanics and Optical Electronics.
2. Look into SU2 and CPX(Basic Exercises), and configure ROOT
3. Solve exercises(Telegrapher's Equation and Spin Evolution of a Pi Pulse) using CPP.
4. Introduction to QASM 2 Programming language.
5. Evaluating Circuits with qiskit, and plotting results.
6. Code a Significant Algorithm of choice(Grover, Shor, Deutsch) etc.

Contents

1	Review of Key Ideas[Optional]	3
1.1	An Introduction To Quantum Computing	3
1.2	Fault Tolerant Quantum Computation: NISQ ERA	4
2	Methodology	6
2.1	Spin Quantum Mechanics	6
2.1.1	Double Split Experiment	6
2.1.2	Stern-Gerlach Experiment	7
2.2	Qubits	8
2.3	Qubit Measurements	10
2.4	Quantum Algorithms	11
2.4.1	Grover’s Algorithm	11
2.4.2	Shor’s Algorithm [Optional]	13
2.4.3	QAOA [Optional]	13
3	Experiments and Results	14
3.1	CPP Experiments	14
3.2	Qiskit Experiments	14
4	Discussion	17
	Bibliography	18

Chapter 1

Review of Key Ideas[Optional]

1.1 An Introduction To Quantum Computing

Quantum Computing[1] is a new and exciting field at the intersection of mathematics, computer science and physics. It concerns a utilization of quantum mechanics to improve efficiency of computation. The central ideas of quantum computation are based on three properties -:

- **Note - :** Associated to any quantum system is a complex vector space known as state space. The state of a closed quantum system is a unit vector in the state space
- **Superposition** - It states that every quantum state can be represented as a sum of two or more distinct states. Mathematically , it refers to a property of solutions to the Schrodinger's equation, since it is linear , any combination of solutons will also be a solution.
- **Entanglement** - It is a property of a multi qubit system and can be thought of as a resource in quantum computing. The key to entanglement is the property that the state space can not be decomposed into component spaces. In summary, by measuring one qubit , it is possible to affect the measurements of other qubits in the system.
- **Interference** - : Interference is the situation where intervention from noise in the environment damages the quantum object, and also the possibility that the wave functions of particles can either reinforce or diminish each other

The paper[1] motivates the use of vectors to describe states and

matrices as ways of describing dynamics, and shows that it is understandable if one looks at a basic toy model. Their model deals with childrens' marbles moving along the edges of a graph, and present many concepts that are core to quantum mechanics.

1.2 Fault Tolerant Quantum Computation: NISQ ERA

The main focus behind doing this part is to explore the questions -:

- **Whether Quantum Computers will have widely used applications ?**
- **When will they be able to solve classical problems faster ?**

When we talk about Quantum Computers we usually mean Fault-tolerant devices. It's usually assumable that these will be able to run, all the algorithms which have been developed over the years and show a potential speed-up in computation as compared to classical computers. However a real Quantum computer with say 50 qubits is not fault tolerant, has some noise in it, and will interact with surroundings which leads to decoherence impeding any computation. One way to process quantum information accurately is by writing Quantum Error Correction codes, but this usually comes with an overhead of more number of qubits and gates.

John Preskill[2] came up with a term (NISQ) to explain this error prone and faulty technology, and presented a few ideas as to how "to accept these flawed machines for now and do something interesting with these devices". NISQ- Noisy Intermediate-Scale-Quantum, 'noisy' because we don't have enough qubits to spare for error correction, and will directly be implemented at the physical layer. 'Intermediate-Quantum' because we have a small number of qubits.

Quantum Computing in NISQ era is here right now, and imperfect gates place severe limits on circuit size. In order to have near-time applications, noise resilience should be a key consideration in quantum circuit design. Nevertheless, technologists have already come up with clever tricks to minimize the amount of quantum

computation through hybrid-algorithms ,like **Variational Quantum Eigen Solvers** and **Quantum Approximate Optimization Algorithms**, which when used with good noise -resilient circuits(based on tensor network constructions improve the prospects of outperforming classical methods. These algorithms work on the principle of combining quantum evaluation of a cost function with a classical feedback loop for seeking a quantum state with a lower value.

Note -: These Hybrid-algorithms might be used , for boosting the power of a classical computer with a quantum computer, but it should not be expected that a quantum computer will solve worst case instances of NP-hard problems, although it might find better approximate solutions, or even find them faster.

To conclude, NISQ - era quantum devices will not be protected by quantum error correction. To reach scalability , we must look for ways to harness more number of physical qubits. Advances in quantum gate fidelity, systems engineering , algorithm design , and error correction protocols can hasten the arrival of the fully fault-tolerant quantum computer. Near-term quantum advantage for useful applications is possible but not guaranteed, one such application which seems like a promising area for quantum advantage is simulating quantum -dynamics of highly entangled systems which is hard of a classical computer. At last, there is a lot of exploration to be which needs to be done in the years to come in areas of Hybrid-classical approaches, Quantum annealing , quantum sensing quantum networking, cryptography etc.

Chapter 2

Methodology

2.1 Spin Quantum Mechanics

Quantum Mechanics is a tool used by Quantum Computers to solve certain computational problems which classical computers cannot. Two essential components of Quantum Mechanics are particles and waves, and each of these have certain properties, which we'll be exploring. **Is light a particle or a wave?, What is spin ?**

We'll answer these questions in this section of the report.

2.1.1 Double Split Experiment

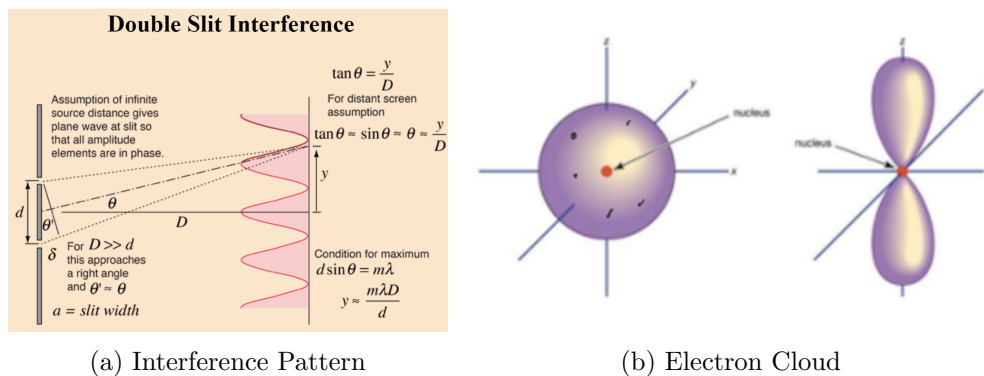


Figure 2.1

Also known as Young's experiment, this involved particle beams or coherent waves passing through two closely-spaced slits, the purpose of which was to measure the resulting impacts on a screen behind them. In quantum mechanics the double-slit experiment demonstrated the inseparability of the wave and particle natures of light and other quantum particles. The inference drawn from this experiment was that particles are not discrete objects and exist in

superposition of states, and that the coherent beams of light were interfering, creating a pattern of bright and dark bands on the screen. To conclude, particles don't have a well defined position, instead they have a wave function given by

$$\psi(x, t)$$

, constrained on position and time. The wavefunction is a probability cloud and it shows how likely it is for a particle to be in a certain region of space at a given time, for example the probability of any particle described by this wavefunction to be at a position x is given by $|\langle x|\psi\rangle|^2$.

Effect of this on Quantum Computing

- Qubits behave like waves
- The superposition of qubit states and interfere with one another
- Using interference we can amplify the probability of the correct answer
- Quantum Interference leads to Quantum speed-up

2.1.2 Stern-Gerlach Experiment

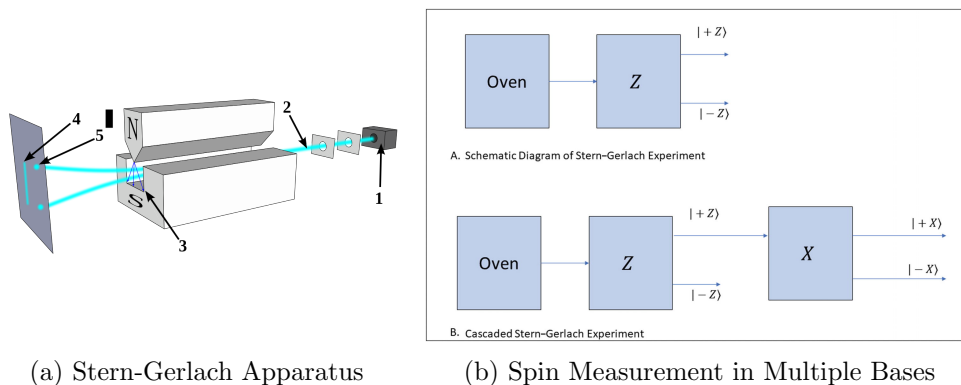


Figure 2.2

This experiment provided the evidence for the existence of qubits in nature. The original experiment (1922) was carried out by sending a beam of hot silver atoms from an oven through a magnetic field, which caused the atoms to be deflected. But since silver's complicated structure obscured the effects of this experiment it was later

carried out with Hydrogen.

Hydrogen contains a proton and an orbiting electron. This electron can be thought of as the electric current around the proton. This current causes the atom to have a magnetic dipole moment. As a result each atom behaves like a bar magnet with an axis corresponding to the axis of the electrons spinning around. Throwing little bar magnets through a magnetic field cause the atoms to be deflected by the field.

How the atom is deflected by the field depends upon both the atom's magnetic dipole moment and the magnetic field generated by the experiment. Two major surprises emerge when this experiment was performed. First, atoms exiting from the Stern-Gerlach device were seen at *discrete* angles. The second most surprising effect was the *number* of peaks seen in the experiment. The Hydrogen atoms being used were such that they should have had zero dipole moment, which would correspond to only one beam of atoms. Instead, two beams were seen, one deflected up by the field and other deflected down.

This puzzle was explained after a considerable effort by introducing a new quantity into nature called the **spin**, which introduced the idea of having a two-level system. Variations of this experiment were carried out to measure the spin in different bases called as the *Cascaded Stern-Gerlach Experiment* Fig 3.2 (b).

Effect of this on Quantum Computing

- All two level systems are equivalent to spin.
- Qubit model describes electron spin.
- Can measure a qubit in different bases.
- This gave rise to the entanglement property.

2.2 Qubits

The backbone for quantum computing is a **Qubit**, which is a two level system, i.e, a system made out of two basis, for example an electron spin. **Note-:** For a system to be Quantum, the following criterion should be met-:

- Small dissipation
- Non linearity
- Isolation from the environment
- Low temperature
- Large vacuum fluctuations

There are many hardware implementations of Qubits , but , for the purpose of this report , **Superconducting Platform** will be of primary interest, as it satisfies majority of the conditions above.

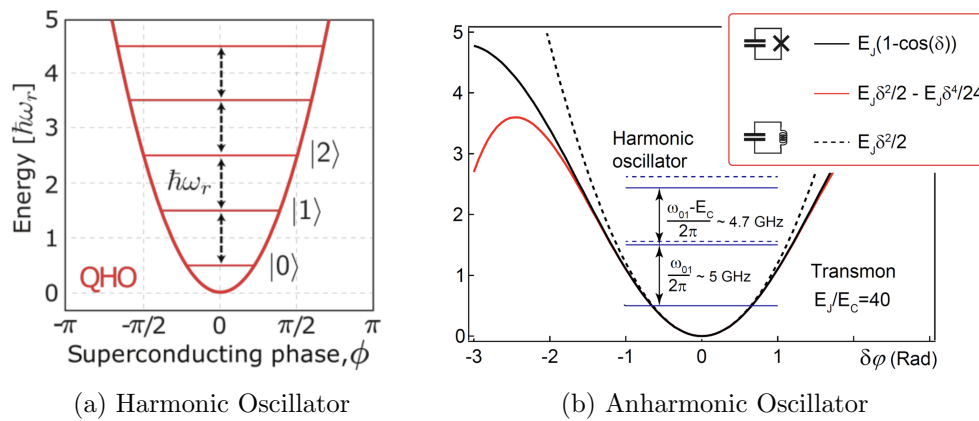


Figure 2.3

What is a Superconductor ?

A superconductor is a material which conducts electricity without any resistance, and as the temperature drops, the material resistance gets lowered. The energy level of such a system is shown in *Fig 3.2(a)*.

If we send a wave at a particular frequency, to excite the atom from the ground state to take it to the first excited state. Now once it is in an excited state and we keep shining light/ wave, it can stimulate the atom to go on to the next level and so on. In essence, there is no way in which we can isolate the energies of the two level system, and can conclude that this system cannot be used as a Qubit. To cater to the problem of Harmonicity , a non-linear inductor(**Josephson's Junction** (*Fig 3.3*) is used so as to discretize the energies of the two level system (*Fig 3.2 (b)*), name $|0\rangle$ and $|1\rangle$. The junction is

basically two superconductors coupled by a thin insulating barrier, through which current *tunnels*.

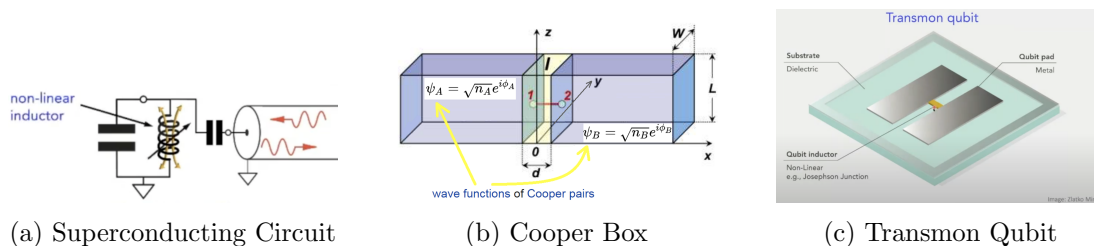


Figure 2.4

Fig 3.3 (a) and (b) are what constitute the **Transmon Qubit**, which can be realised as a persistent current in the superconducting circuit. It is anharmonically engineered, immune to V_g variations and a phase state qubit.

2.3 Qubit Measurements

In this section, an investigation on Quantum Hardware using Microwave pulses was performed. Results are shown in the Experiment's section of this report.

- **Qubit Frequency Scan - :** The frequency of qubit at which it becomes resonant is the energy difference between its ground and excited states $|0\rangle$, $|1\rangle$. Though superconducting qubits exhibit a higher number of energy levels. They're engineered for non linearity to keep higher levels out of reach
- **Calibrating π pulse using a rabi experiment -:** Once we know the frequency of our qubit, the next step is to determine the strength of a π pulse. It is the pulse which takes the qubit from $|0\rangle$ to $|1\rangle$ and vice versa.
- **Discriminating 0 vs 1 - :** Once we have the calibrated π pulse, we can use it to find what the states $|0\rangle$ and $|1\rangle$ look like in our measurements, by repeatedly preparing and plotting the measured signal.
- **T1 determination - :** The time a qubit takes to decay from the excited state to the ground state, is known as the T1 time. We first apply a π pulse followed by a certain delay in time

and then apply a measure pulse. Then we vary and re-run for a number of delays.

- **Precise frequency measurement with a Ramsey Experiment** - : This is done using a Ramsey pulse sequence. In this pulse sequence, we first apply a $\pi/2$ pulse, wait some time δt and then apply another $\pi/2$ pulse.
- **Measuring T2 using Hahn Echoes** - : Measure the coherence time of our qubit. We follow a Ramsey experiment and add a π pulse between two $\pi/2$ pulses. The π pulse at time τ reverses the accumulation of phase, and results in an echo at time 2τ where we apply the last $\pi/2$ pulse to do our measurement.
- **Dynamical Decoupling** - : It allows us to cancel different frequencies of noise and is used to extract longer coherence times from qubits.

2.4 Quantum Algorithms

2.4.1 Grover's Algorithm

Grover's Algorithm is one of those quantum algorithms which shows a potential advantage over classical algorithms. It can speed up an unstructured search quadratically, and uses extend beyond that. It can speed up in tasks like Optimal Scheduling , estimation of mean and median of some distribution, element counting and in general NP complete algorithms which generally contain exhaustive search as a subroutine can be sped up as well(**Travelling Salesman Problem**).

The algorithm owes everything to just one trick known as the **Amplitude Amplification**[3]. Which in essence leverages quantum interference to constructively interfere with the correct solution and destructively interfere with the incorrect solution. The main steps of the algorithm are as follows:-

1. Start by assigning equal probability to every cell being the right answer. This we do by applying superposition to all states.
2. Use amplitude amplification trick to maximize the probability

of measuring the right state. This step majorly consists of two transformations-:

$$|\psi_i\rangle \xrightarrow{-\mathbf{U}_s\mathbf{U}_\omega} |\psi_{i+1}\rangle$$

where $|\omega\rangle$ is the target state and

$|s\rangle$ the superposition state

$$\mathbf{U}_s = \mathbf{2} - |s\rangle\langle s|$$

$$\mathbf{U}_\omega = \mathbf{2} - |\omega\rangle\langle\omega|$$

2.1 We have applied the oracle reflection to the winner state here, which particularly corresponds to a reflection of state $|s\rangle$ about the state $|s'\rangle$. This transformation adds a negative phase to the winner state which in turn lowers the average amplitude.

2.2 The next step is called inversion about the mean. This is done so as to mark the winner state with the maximum probability. These transformations together account for one rotation. This is also known as the **diffusion operator**.

$$U_s = 2|s\rangle\langle s| - \mathbf{1}.$$

Note:- While implementing this part of the circuit the first reflection is around $|s'\rangle$ followed by the second reflection about $|s\rangle$. Since the second reflection is about $|s\rangle$, we want to add a negative phase to every state orthogonal to it. However it is usually not possible to directly reflect across $|s\rangle$ because all the states are in superposition. So first we transform it to the $|0\rangle$ state by applying a **Hadamard** gate and then applying a negative phase to states orthogonal this new state. At last, we apply a **Hadamard** again to get back our original state, with the winner standing out.

3. Measure all the states.

This second step of the circuit is usually known as the **Grover**

Iteration and we apply it $O\sqrt{N}$ times, where N is the number of qubits. Although this algorithm proves to beat any classical counterpart, it has some shortcomings-:

- Not enough qubits to encode a large enough dataset in the **QRAM**[4].
- Qubits are Noisy with smaller **Decoherence and Relaxation Times**
- **Gate fidelity** decreases with number of Qubits.

Since this is something beyond the scope of this project I'd briefly introduce the key ideas of these algorithms -:

2.4.2 Shor's Algorithm [Optional]

It[5] is used to factor the product of two primes faster than any classical computer, by representing a factoring problem as a phase estimation problem. Here are two important concepts utilized for solving this-:

- *Quantum Fourier Transform* - : The quantum Fourier transform (QFT) is the quantum implementation of the discrete Fourier transform over the amplitudes of a wavefunction.
- *Quantum Phase Estimation* - : Given a unitary U , the algorithm estimates θ in $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$. Here $|\psi\rangle$ is an eigenvector and $e^{2\pi i\theta}$ is the corresponding eigenvalue.

2.4.3 QAOA [Optional]

QAOA[6] is one such circuit which extends on the idea of utilising NISQ devices for real applications and is used for combinatorial optimization(TSP/Nurse Scheduling Problem). Following are the steps of a QAOA circuit-:

- Map the classical cost function to a Hamiltonian H .
- Apply an entangling unitary followed by a single qubit unitary with tunable parameters
- perform measurement

Chapter 3

Experiments and Results

3.1 CPP Experiments

Since complex numbers, vectors and matrices are one of the fundamental tools in **Quantum Computing**, we used 2 cpp packages namely CPX and SU(2) for our work. All the code for this section was tested on **HYBRILit Supercomputer**.

- Here we perform a sample exercise on CPX , simulating some algebra on Complex and Real Numbers.
- An exercise simulating Vectors and Matrices on SU2.
- We proceeded on to some advanced exercises and utilized **HYBRILit's** compute power to solve for telegrapher's equation and spin evolution of a pi-pulse.

3.2 Qiskit Experiments

- **Freq Scan - :** The valley near the centre corresponds to the location of the qubit frequency.
- **Rabi Experiment - :** When we change the drive amplitude in small increments and measure the state of qubit each time we see these oscillations commonly known as Rabi oscillations.
- **0/1 Discriminator - :** We see two populations of $|0\rangle$ and $|1\rangle$ form their own clusters.
- **T1 and T2 - :** Here we have plotted signal vs delay time.
- **Dynamic Decoupling - :** measured signal vs total time before measurement.

```
[waves@space16 ETX]$ make run
export LD_LIBRARY_PATH=/zfs/hybrilit.jinr.ru/user/w/waves/libs/GCX/lib:lib.../CPX/lib; \
./test

-----

cpx<int> + <real> = 1+i
Multiplication of Two Complex Numbers (1 + i)*(2 -5i) = 12-i
cpx<int> + <real> = (1+i).real

cpx<int> + cpx<dbl> = 2i
cpx<int> + cpx<dbl> = 2i.dblx

sqrt(1+i*1.0e-20) = 1
sqrt(1-i*1.0e-20) = -1

sqrt(k) = 0.7071+0.7071i
exp(i*pi/4) = 0.7071+0.7071i
exp(i*2pi) = 1.real

4*log(sqrt2+i*sqrt2) = 3.1415927i.real

sqrt(-1) = -nan
tough luck !... -1 = arithmetic, not cpx
try with cpx !

sqrt(-1) = i.dblx

Complex Phase [0...2pi] 5.4977871

Euler's Formula exp(i*phi) = cos(phi) + isin(phi)
Enter complex number -: 1
Enter phase phi -: 3.14159265358979323846264L
exp(i*phi) = -1.real
cos(phi) + isin(phi) = -1

-----

[waves@space16 ETX]$ █
```

(a) Complex Numbers

```
[waves@space19 ETX]$ make run
export LD_LIBRARY_PATH=/zfs/hybrilit.jinr.ru/user/w/waves/libs/GCX/lib:lib.../PSI/lib:lib.../SU2/lib; \
./test

su2<cpx<int>> + <real> = [2 + sy].cpx<real>
su2<cpx<int>> + <real> = 1 + sy
sqrt(sx) = 0.5+0.5i + (0.5-0.5i)sx
log(sx) = 1.57881 - 1.57881ix
exp(j*pi/2*sx) = isx

Some State on the bloch sphere(1+i, 1-i)

Applying Hademard to the |1> state
1/sqrt(2)*(1, -1)

Applying Hademard to the |0> state
1/sqrt(2)*(1, 1)

-2i

Quantum Average of Y rotation 0

-----

[waves@space19 ETX]$
```

(b) Vectors and Matrices

```
auto propagator(real x, real t, real f)
{
  real gamma = sqrt(1+f*f*Ld*Ld/c/c)
  real beta = sqrt(gamma*gamma-1) / gamma ;
  return e^(-(1+sx*beta)*(j*sy)*(x-beta*c*t) ;
             *gamma*gamma/Ld) ;}
}
```

(c) Telegrapher's Code

$$\partial_x + \sigma_1 \partial_{ct} + j\sigma_2 \lambda_d^{-1} = 0 |_{\phi}$$

$$\phi = e^{-\gamma^2(1+\sigma_1\beta)\frac{j\sigma_2}{\lambda_d}(x-vt)} |_{\phi_0}$$

(d) Telegrapher's Equation

Figure 3.1

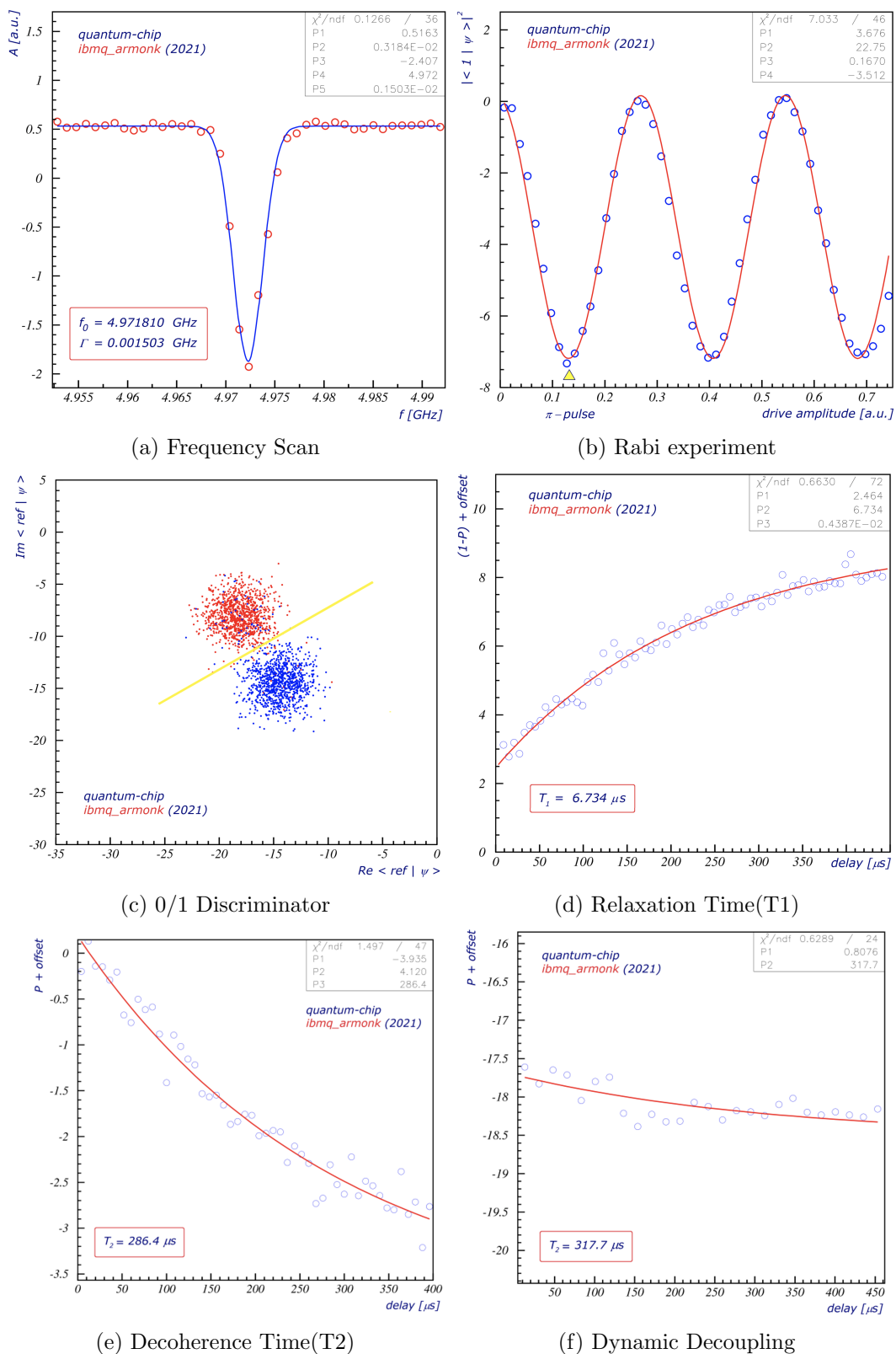


Figure 3.2

Chapter 4

Discussion

This program helped me gain essential knowledge about the underlying Quantum mechanics and the hardware side of Quantum Computing. Working with a supercomputer was a great experience in itself. I also explored those aspects of this field which otherwise i would never have stepped into, especially the part about Superconducting qubits. Here are some of the things i came across while pursuing my own research within the project phase-:

- Quantum Machine Learning - PennyLane's API, Quantum Neural Networks, Barren Plateaus.
- Variational Quantum Eigen Solver and QAOA- Qiskit textbook
- Othe hardware implementations of qubits like - Trapped Ion, Diamon NV centres and Photonics

Bibliography

- [1] N. S. Yanofsky, “An introduction to quantum computing,” 2007.
- [2] J. Preskill, “Quantum computing in the nisq era and beyond,” 2018.
- [3] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, “Quantum amplitude amplification and estimation,” 2000.
- [4] V. Giovannetti, S. Lloyd, and L. Maccone, “Quantum random access memory,” 2007.
- [5] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” 1995.
- [6] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” 2014.